

**LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB & MOBILE I**



NAMA : HUSSAIN SYACH NURALAM
NIM : 193020503029
KELAS : A
MODUL : FORM HANDLING

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021**

BAB I

PENDAHULUAN

1.1 Tujuan

Tujuan dari dilaksanakannya praktikum Pemrograman Web dan Mobile, modul II tentang Form Handling adalah sebagai berikut.

- 1) Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML.
- 2) Mahasiswa mampu membuat batasan-batasan untuk menangani inputan dari form HTML.

1.2 Landasan Teori

1.2.1 Pengertian Form Handling

Hal paling penting untuk diperhatikan ketika berhadapan dengan HTML dan PHP adalah bahwa setiap elemen form dalam sebuah halaman HTML secara otomatis akan tersedia untuk skrip PHP yang akan digunakan. Form handling ialah suatu mekanisme untuk menangani suatu masukan dari form yang dikirim oleh pengguna. Form handling berhubungan dengan metode yang dikirim dan bagaimana menangani pengiriman data berdasarkan metode yang digunakan (Adhitya, 2012).

Dalam aplikasi web ada kalanya memerlukan inputan atau masukan dari pengguna. Untuk melakukannya dapat memanfaatkan form HTML. Form HTML terdiri dari tag-tag seperti <form>, <input>, <textarea> dan <select> yang mana penggunaannya disesuaikan berdasarkan data yang akan dikoleksi (Hasudungan, 2018). Form HTML dibentuk menggunakan tag, dan tag form memiliki attribute method yang digunakan untuk menentukan bagaimana cara form mengirim data ke server (data dikirim ke halaman yang diset di dalam attribute action).

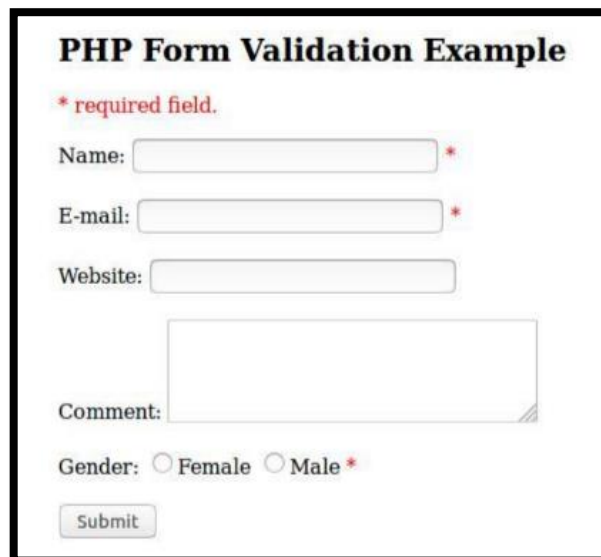
Variabel superglobal PHP `$_GET` dan `$_POST` digunakan untuk mengumpulkan data-form. `$_POST` termasuk dalam kategori Predefined Variable yang disediakan oleh library PHP. Kegunaan dari `$_POST` hampir sama dengan `$_GET` yaitu digunakan untuk mengambil suatu nilai yang dikirimkan oleh form bedanya `$_POST` mengambil nilai yang dikirim oleh form dengan method="post". Semua nilai / informasi yang dikirimkan oleh form yang menggunakan method POST tidak akan terlihat oleh user yang mengakses, dikarenakan informasi yang dikirim akan tidak ditampilkan di Address Bar Web Browser. Selain `$_POST` juga tidak memiliki batasan pada jumlah informasi yang dikirim.

GET dan POST membuat sebuah array (contoh array(kunci => nilai, kunci2 => nilai2, kunci3 => nilai3, ...)). Array ini menyimpan pasangan kunci/nilai, dimana kunci kunci adalah nama-nama dari form control dan nilai-nilai adalah data input dari user. Method GET diakses menggunakan `$_GET` dan method POST diakses menggunakan `$_POST`. Kedua variabel ini adalah variabel superglobal, yang selalu bisa diakses, tanpa memperhatikan lingkup dan bisa diakses dari fungsi, class atau file yang berbeda tanpa harus melakukan teknik khusus. `$_GET` adalah sebuah array dari variabel yang dikirimkan ke skrip melalui parameter URL. `$_POST` adalah sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST.

Informasi dikirim dari sebuah form dengan method GET bisa dilihat oleh semua orang (semua nama dan nilai variabel ditampilkan di URL). GET juga memiliki batas pada jumlah informasi yang dikirim. Batasannya adalah sekitar 2000 karakter. Namun, karena variabel ditunjukkan di URL, ia memungkinkan untuk dilakukan bookmark halaman. Dalam beberapa kasus, hal ini sangat bermanfaat. GET bisa digunakan untuk mengirimkan data yang tidak sensitif.

Informasi yang dikirim dari sebuah form dengan method POST tidak bisa dilihat oleh siapapun (semua nama-nama atau nilai-nilai tertanam didalam body request HTTP) dan tidak memiliki batasan jumlah informasi yang akan dikirim. POST juga mendukung fungsionalitas lanjutan seperti dukungan untuk input biner multi-part ketika sedang melakukan upload file ke server. Namun, karena variabel tidak ditampilkan di URL, tidak mungkin untuk dilakukan bookmark halaman (data tidak ter-bookmark). Developer lebih baik menggunakan POST untuk mengirimkan data form (Dosen Teknik Informatika, 2021).

Berikut adalah contoh dari validasi form PHP.



PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: ☐ Female ☐ Male *

Gambar 1.1 Contoh Form PHP

Form HTML yang akan kita gunakan pada modul ini, mengandung bermacam macam field input, misalnya text field yang harus diisi dan text field yang opsional, tombol pilihan (radio button), dan tombol submit. Rule atau aturan validasi untuk form diatas adalah sebagai berikut:

- 1) Name Dibutuhkan. + Harus hanya mengandung huruf dan spasi
- 2) E-mail Dibutuhkan. + Harus mengandung sebuah alamat email yang valid dengan @ dan .
- 3) Website Opsional. Jika ada, harus mengandung URL yang valid.
- 4) Comment Opsional. Field input multi-line (text area).
- 5) Gender Dibutuhkan. Harus memilih salah satu

Kode HTML untuk membentuk Form tersebut adalah sebagai berikut:

1) Text Field

Field nama, email dan website adalah elemen-elemen text input, dan field komentar adalah textarea yaitu sebagai berikut:

- 1.1) Name: `<input type="text" name="name">` E-mail: `<input type="text" name="email">`
- 1.2) Website: `<input type="text" name="website">`
- 1.3) Comment: `<textarea name="comment" rows="5" cols="40"></textarea>`

2) Radio Button

Field jenis kelamin adalah radio button yaitu sebagai berikut:

- 2.1) `<input type="radio" name="gender" value="female">Female`
- 2.2) `<input type="radio" name="gender" value="male">Male`

3) Form Element

Kode HTML untuk membentuk form pada gambar diatas adalah sebagai berikut:

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);? >">
```

Ketika form disubmit, data pada form dikirim dengan method "post". `$_SERVER["PHP_SELF"]` adalah variabel super global yang mengembalikan nama file dari skrip yang sedang dieksekusi. Sehingga kode form diatas mengirim data pada form ke halaman itu

sendiri. Sedangkan fungsi `htmlspecialchars()` adalah fungsi yang mengkonversikan karakter-karakter spesial ke entitas HTML. Sebagai contoh, fungsi tersebut akan mengkonversikan karakter `<` dan `>` menjadi `<` dan `>`. Fungsi ini mencegah injeksi yang bisa dilakukan dengan HTML atau javascript (Cross-site Scripting Attack) pada form tersebut.

Variabel `$_SERVER["PHP_SELF"]` bisa digunakan oleh hacker! Jika `PHP_SELF` digunakan pada halaman web, user bisa memasukkan skrip dengan terlebih dahulu memasukkan garis miring (`/`) kemudian beberapa perintah Cross Site Scripting (XSS) untuk dieksekusi. XSS adalah tipe kelemahan keamanan komputer yang secara tipikal ditemukan dalam aplikasi web.

Untuk menghindari penyalahgunaan `$_SERVER["PHP_SELF"]` adalah dengan menggunakan fungsi `htmlspecialchars()`. Fungsi tersebut akan mengkonversikan karakter khusus ke entitas HTML. Ketika user memasukkan URL dengan tag script seperti contoh sebelumnya, maka akan ditranslasikan sebagai berikut:

```
<form method="post"
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

Gambar 1.2 URL untuk menghindari penyalahgunaan `$_SERVER`

Untuk mengvalidasi data form dengan PHP, hal pertama yang harus dilakukan adalah memasukkan semua variabel melalui fungsi `htmlspecialchars()`. Kemudian ada juga dua hal ketika user melakukan submit form:

- 1) Membuang karakter-karakter yang tidak dibutuhkan (seperti spasi extra, tab extra, dan baris baru yang ekstra) dari data input user (dengan fungsi `trim()`).
- 2) Membuang backslash (`\`) satu garis miring dari data input user (dengan fungsi `stripslashes()`).

Langkah berikutnya adalah membuat fungsi yang akan melakukan pemeriksaan kebenaran data yang diinputkan oleh user. Hal pada permulaan skrip, adalah pemeriksaan apakah form sudah disubmit menggunakan `$_SERVER["REQUEST_METHOD"]`. Jika `REQUEST_METHOD` adalah `POST`, maka form telah disubmit dan seharusnya tervalidasi. Jika belum tersubmit, lewati langkah validasi dan tampilkan form kosong.

BAB II

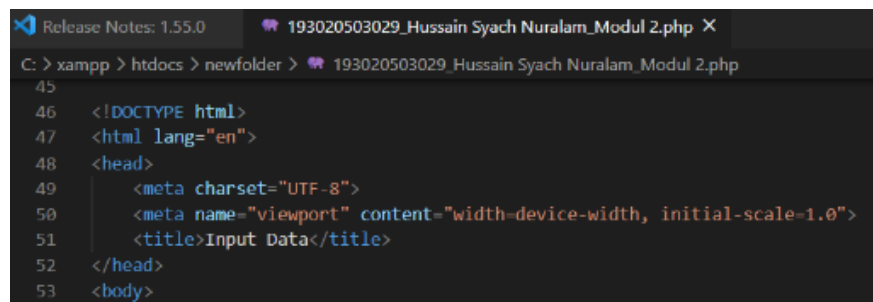
PEMBAHASAN

2.1 Langkah Kerja

Buatlah program web untuk menginputkan username dan password menggunakan form dan penanganan input data dengan kriteria sebagai berikut:

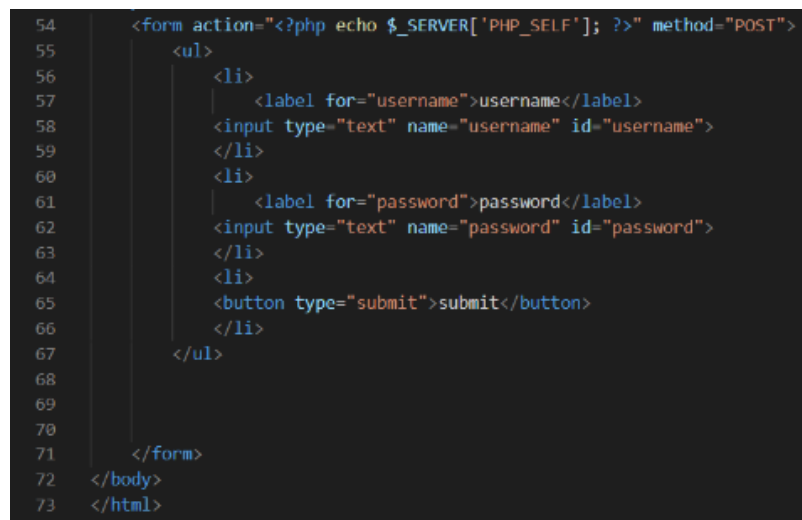
- 1) Username yang diinputkan tidak boleh lebih dari tujuh karakter.
- 2) Password yang diinputkan harus terdiri dari huruf kapital, huruf kecil, angka dan karakter khusus.
- 3) Jumlah karakter password tidak boleh kurang dari sepuluh karakter.

2.2 Pembahasan



```
45
46 <!DOCTYPE html>
47 <html lang="en">
48 <head>
49   <meta charset="UTF-8">
50   <meta name="viewport" content="width=device-width, initial-scale=1.0">
51   <title>Input Data</title>
52 </head>
53 <body>
```

Gambar 2.1 Sintaks Dasar Program Input Data



```
54 <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
55   <ul>
56     <li>
57       <label for="username">username</label>
58       <input type="text" name="username" id="username">
59     </li>
60     <li>
61       <label for="password">password</label>
62       <input type="text" name="password" id="password">
63     </li>
64     <li>
65       <button type="submit">submit</button>
66     </li>
67   </ul>
68 </form>
69
70
71 </body>
72 </html>
```

Gambar 2.2 Sintaks Dasar Program Input Data

Dari gambar-gambar di atas, terdapat beberapa sintaks dasar yang harus dimasukkan di dalam program PHP agar program tersebut dapat dieksekusi. Berikut adalah penjelasannya.

- 1) `<!DOCTYPE html>;` adalah suatu deklarasi yang digunakan untuk mengidentifikasi jenis dokumen HTML yang digunakan sehingga browser dapat menentukan bagaimana memperlakukan kode tersebut.
- 2) `<html>;` adalah tag utama yang digunakan untuk memulai dokumen HTML.
- 3) `<head>;` adalah tag utama yang digunakan untuk membuat bagian kepala dokumen.
- 4) `<title>;` adalah tag utama yang digunakan untuk mengatur judul web.
- 5) `<body>;` adalah tag utama yang digunakan untuk membuat bagian tubuh dokumen (Muhardian, 2020).
- 6) `` menunjukkan sebuah Unordered List atau Daftar Acak. HTML `` element digunakan untuk merepresentasikan sebuah daftar item atau koleksi beberapa item yang dapat disusun kembali secara acak tanpa penomoran.
- 7) Untuk menulis sebuah item dalam daftar (list), digunakan element `` yang menunjukkan List Item (Fadul, 2018).
- 8) Meta tag adalah “data tentang data”, dimana tag ini ditujukan bukan kepada user, tetapi kepada web browser atau kepada “robot program” seperti mesin pencari (search engine) (Andre, 2014).
- 9) `<form>` element digunakan untuk menampung macam-macam element yang berkaitan dengan sebuah form. Dalam sebuah form biasanya terdapat kotak input (control) dan element lainnya yang dapat diedit kemudian ditulis untuk dikirim pada sebuah server untuk diproses guna mendapatkan informasi tertentu dari atau untuk user (Fadul, 2018).

Sebelum diberikan kode sintaks dasar yang menghubungkan antara HTML dan PHP, diberikanlah kode-kode program yang mencakup variabel-variabel dan percabangan if seperti yang ditunjukkan pada gambar-gambar berikut.

```
C: > xampp > htdocs > newfolder > 193020503029_Hussain Syach Nuralam_Modul 2.php
1  <?php
2      if ($_SERVER["REQUEST_METHOD"] == "POST"){
3          $username = $_REQUEST["username"];
4          $password = $_REQUEST["password"];
5          $user = strlen($username);
6          $pass = strlen($password);
7          $x = false;
8
9          if($user>7){
10             echo "username lebih dari 7 karakter atau huruf<br>";
11             $x = true;
12         }
13
14         if (lpeg_match("/[A-Z]/", $password) ) {
15             echo "password dengan huruf kapital<br>";
16             $x = true;
17         }
```

Gambar 2.3 Sintaks Percabangan Program Input Data Serta Penggunaan Handling

Sebelum masuk pada penggunaan pernyataan if, terdapat sintaks <?php. sintaks ini berfungsi sebagai kode wajib dalam program PHP agar program dapat dijalankan. Di bawahnya, terdapat percabangan if yang mana terdapat handling di dalamnya yakni \$_SERVER yang mana fungsinya nanti akan menghasilkan value true atau false. Hal ini menjadikan \$_SERVER seolah-olah memiliki kegunaan yang sama dengan fungsi isset(\$_POST) dikarenakan.

\$_REQUEST adalah variabel superglobal yang digunakan untuk mengumpulkan data setelah menyerahkan sebuah HTML form. Penggunaanya pada \$username dan \$password adalah untuk mengumpulkan data username dan password dari HTML form yang telah disubmit. Fungsi strlen digunakan untuk menghitung panjang karakter yang ada di dalam string yang dalam kasus ini adalah data dari \$username dan \$password.
 merupakan element yang ada pada html. Elemen ini berfungsi untuk menunjukkan sebuah line break (garis baru) (Fadullah, 2018). Selain itu, variabel \$x berguna untuk memberikan pernyataan yang salah (false).

Kemudian, penggunaan percabangan `if` pada sintaks di atas digunakan agar variabel `$user` dan `$password`. `Preg_match` adalah salah satu dari regular expression, yang mana regular Expression atau biasa disingkat `regex` adalah suatu metode untuk mengenali atau mendeteksi suatu pola tertentu pada suatu string. `Regex` berfungsi untuk mendeteksi pola string seperti email, hashtag, link dan pola-pola kompleks lainnya dengan hanya satu ekspresi saja (Huda, 2020).

Pada pernyataan `if($user>7)`, ini digunakan agar username yang dimasukan dan dapat disubmit bukanlah data user yang memiliki username dengan lebih dari 7 karakter di dalamnya. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik. Penggunaan `$x = true` digunakan agar jika hasil dari username adalah dengan menggunakan lebih dari 7 karakter, maka `$x` yang bernilai `true` akan mencegah data tersebut dapat disubmit karena tidak sesuai dengan aturan. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik.

Pada `if (!preg_match("/[A-Z]/", $password))`, ini digunakan agar `regex` dapat mengenali atau mendeteksi pola string yang dalam kasus ini adalah A-Z. jika tidak menggunakan karakter A-Z dalam password, maka `$x = true` akan mencegah data tersebut dapat disubmit karena tidak sesuai dengan aturan. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik.

```

19  if (!preg_match("/[a-z]/", $password)) {
20      echo "password dengan huruf kecil\n";
21      $x = true;
22  }
23
24  if (!preg_match("/[^a-zA-Z\d]/", $password)) {
25      echo "password dengan karakter spesial\n";
26      $x = true;
27  }
28
29  if (!preg_match("/[0-9]/", $password)) {
30      echo "password digit\n";
31      $x = true;
32  }
33
34  if($pass>10){
35      echo "password lebih dari 10 karakter";
36      $x = true;
37  }

```

Gambar 2.4 Sintaks Percabangan Program Input Data Serta Penggunaan Handling

Pada `if (!preg_match("/[a-z]/", $password))`, ini digunakan agar regex dapat mengenali atau mendeteksi pola string yang dalam kasus ini adalah a-z. Jika tidak menggunakan karakter a-z dalam password, maka `$x = true` akan mencegah data tersebut dapat disubmit karena tidak sesuai dengan aturan. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik.

Pada `if (!preg_match("/[^a-zA-Z\d]/", $password))`, ini digunakan agar regex dapat mengenali atau mendeteksi pola string yang dalam kasus ini adalah karakter-karakter special yang mencakup `[^a-zA-Z\d]`. Jika tidak menggunakan karakter `[^a-zA-Z\d]` dalam password, maka `$x = true` akan mencegah data tersebut dapat disubmit karena tidak sesuai dengan aturan. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik. Penggunaan meta karakter pada regex terdapat pada gambar di atas. Berikut adalah penggunaan meta karakternya.

- 1) `[^a-zA-Z]` merupakan meta yang digunakan dalam regex. Yang mana, arti dari meta ini adalah pola yang tidak terdiri dari set karakter di dalam kurung siku.
- 2) `\d` merupakan meta yang digunakan dalam regex. Yang mana, arti dari meta ini adalah angka digit (0-9) (Huda, 2020).

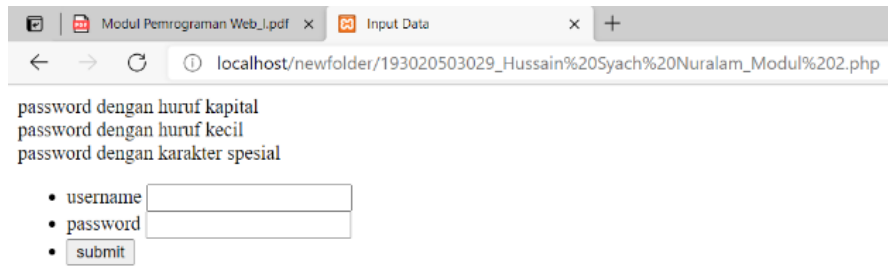
Pada `if (!preg_match("/[0-9]/", $password))`, ini digunakan agar regex dapat mengenali atau mendeteksi pola string yang dalam kasus ini adalah 0-9. Jika tidak menggunakan karakter 0-9 dalam password, maka `$x = true` akan mencegah data tersebut dapat disubmit karena tidak sesuai dengan aturan. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik.

Pada pernyataan `if($pass>10)`, ini digunakan agar username yang dimasukan dan dapat disubmit bukanlah data user yang memiliki username dengan lebih dari 10 karakter di dalamnya. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik. Penggunaan `$x = true` digunakan agar jika hasil dari username adalah dengan menggunakan lebih dari 10 karakter, maka `$x` yang bernilai `true` akan mencegah data tersebut dapat disubmit karena tidak sesuai dengan aturan. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik.

```
39         if( $x == false ){  
40             echo "berhasil";  
41         }  
42     }  
43  
44     ?>
```

Gambar 2.5 Sintaks Percabangan Program Input Data Serta Penggunaan Handling

Pada pernyataan `if ($x == false)`, dapat dipastikan apabila data-data berupa username dan password dimasukan sesuai dengan syarat maka nilai `$x = false` sehingga data-data tersebut dapat disubmit. Penggunaan `echo` telah dijelaskan untuk menampilkan teks yang berada di dalam tanda petik. Tag `?>` digunakan sebagai kode untuk mengakhiri PHP dan wajib digunakan saat digabung dengan bahasa pemrograman lain seperti HTML. Sehingga didapat hasil output dari program input data diri yang ditunjukan sebagai berikut.



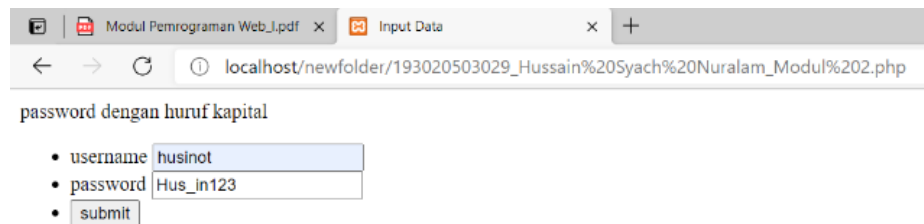
Modul Pemrograman Web_I.pdf x Input Data x +

localhost/newfolder/193020503029_Hussain%20Syach%20Nuralam_Modul%202.php

password dengan huruf kapital
password dengan huruf kecil
password dengan karakter spesial

- username
- password
-

Gambar 2.6 Program Gagal Submit Karena Tidak Sesuai



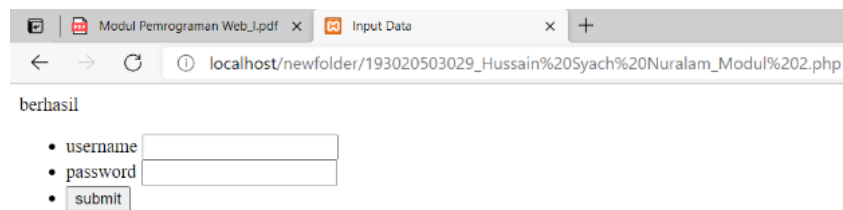
Modul Pemrograman Web_I.pdf x Input Data x +

localhost/newfolder/193020503029_Hussain%20Syach%20Nuralam_Modul%202.php

password dengan huruf kapital

- username
- password
-

Gambar 2.7 Pengisian Data Username dan Password



Modul Pemrograman Web_I.pdf x Input Data x +

localhost/newfolder/193020503029_Hussain%20Syach%20Nuralam_Modul%202.php

berhasil

- username
- password
-

Gambar 2.8 Program Berhasil Submit

BAB III

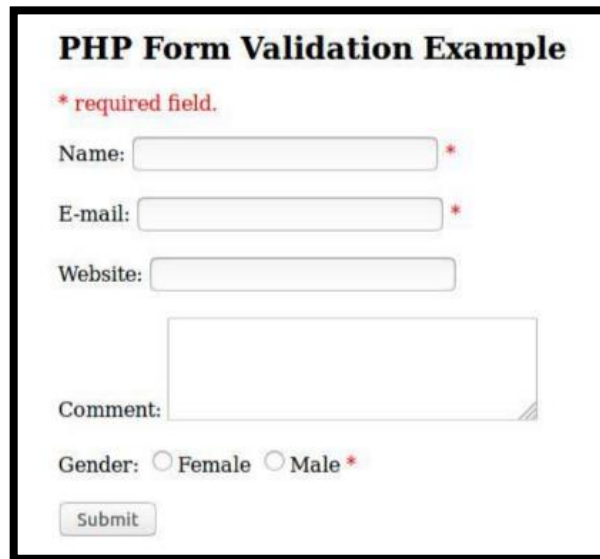
KESIMPULAN

Form handling ialah suatu mekanisme untuk menangani suatu masukan dari form yang dikirim oleh pengguna. Form handling berhubungan dengan metode yang dikirim dan bagaimana menangani pengiriman data berdasarkan metode yang digunakan. Dalam aplikasi web ada kalanya memerlukan inputan atau masukan dari pengguna. Untuk melakukannya dapat memanfaatkan form HTML. Form HTML terdiri dari tag-tag seperti <form>, <input>, <textarea> dan <select> yang mana penggunaannya disesuaikan berdasarkan data yang akan dikoleksi. Form HTML dibentuk menggunakan tag, dan tag form memiliki attribute method yang digunakan untuk menentukan bagaimana cara form mengirim data ke server (data dikirim ke halaman yang diset di dalam attribute action).

DAFTAR PUSTAKA

- Adhitya, A. (2012). *PHP Form Handling*. <https://Www.Lab-Informatika.Com/>.
<https://www.lab-informatika.com/php-form-handling>
- Andre. (2014). *Pengertian Meta tag Charset UTF-8 pada HTML5*.
<https://Www.Duniaikom.Com/>. <https://www.duniaikom.com/tutorial-belajar-html5-pengertian-meta-tag-charset-utf-8-pada-html5/>
- Dosen Teknik Informatika. (2021). *MODUL PRAKTIKUM PEMROGRAMAN WEB I. 1*, 60.
- Fadul, F. (2018a). *HTML form tag*. <https://Www.Apacara.Com/>.
<https://www.apacara.com/tutorial/html/html-form-tag.html>
- Fadul, F. (2018b). *HTML ul tag*. <https://Www.Apacara.Com/>.
<https://www.apacara.com/tutorial/html/html-ul-tag.html>
- Fadullah. (2018). *HTML br tag. Belajar br element*. [Apacara.Com](https://www.apacara.com/tutorial/html/html-br-tag.html).
<https://www.apacara.com/tutorial/html/html-br-tag.html>
- Hasudungan, R. (2018). *BAB 8 FORM HANDLING DAN VALIDASI* (Vol. 1).
- Huda, N. (2020). *PHP Belajar Regular Expression*. <https://Jagongoding.Com/>.
<https://jagongoding.com/web/php/menengah/regular-expression/>
- Muhardian, A. (2020). *Belajar HTML #02 Apa itu Tag, Elemen, dan Atribut dalam HTML*. [Petanikode.Com](https://www.petanikode.com/html-tag-elemen-atribut/). <https://www.petanikode.com/html-tag-elemen-atribut/>

LAMPIRAN



PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

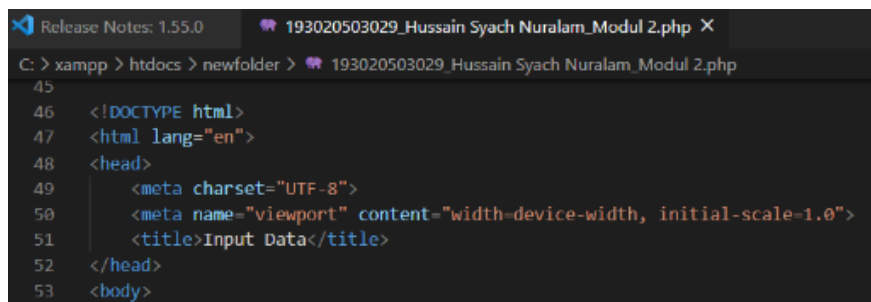
Comment:

Gender: ☐ Female ☐ Male *

Gambar 1.1 Contoh Form PHP

```
<form method="post"
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

Gambar 1.2 URL untuk menghindari penyalahgunaan \$_SERVER



```
45
46 <!DOCTYPE html>
47 <html lang="en">
48 <head>
49   <meta charset="UTF-8">
50   <meta name="viewport" content="width=device-width, initial-scale=1.0">
51   <title>Input Data</title>
52 </head>
53 <body>
```

Gambar 2.1 Sintaks Dasar Program Input Data

```

54     <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
55         <ul>
56             <li>
57                 <label for="username">username</label>
58                 <input type="text" name="username" id="username">
59             </li>
60             <li>
61                 <label for="password">password</label>
62                 <input type="text" name="password" id="password">
63             </li>
64             <li>
65                 <button type="submit">submit</button>
66             </li>
67         </ul>
68     </form>
69
70
71 </body>
72 </html>
73

```

Gambar 2.2 Sintaks Dasar Program Input Data

```

C: > xampp > htdocs > newfolder > 193020503029_Hussain Syach Nuralam_Modul 2.php
1  <?php
2      if ($_SERVER["REQUEST_METHOD"] == "POST"){
3          $username = $_REQUEST["username"];
4          $password = $_REQUEST["password"];
5          $user = strlen($username);
6          $pass = strlen($password);
7          $x = false;
8
9          if($user>7){
10             echo "username lebih dari 7 karakter atau huruf<br>";
11             $x = true;
12         }
13
14         if (!preg_match("/[A-Z]/", $password) ) {
15             echo "password dengan huruf kapital<br>";
16             $x = true;
17         }

```

Gambar 2.3 Sintaks Percabangan Program Input Data Serta Penggunaan Handling

```

19     if (!preg_match("/[a-z]/", $password)) {
20         echo "password dengan huruf kecil\n";
21         $x = true;
22     }
23
24     if (!preg_match("/[^a-zA-Z\d]/", $password)) {
25         echo "password dengan karakter spesial\n";
26         $x = true;
27     }
28
29     if (!preg_match("/[0-9]/", $password)) {
30         echo "password digit\n";
31         $x = true;
32     }
33
34     if($pass>10){
35         echo "password lebih dari 10 karakter";
36         $x = true;
37     }

```

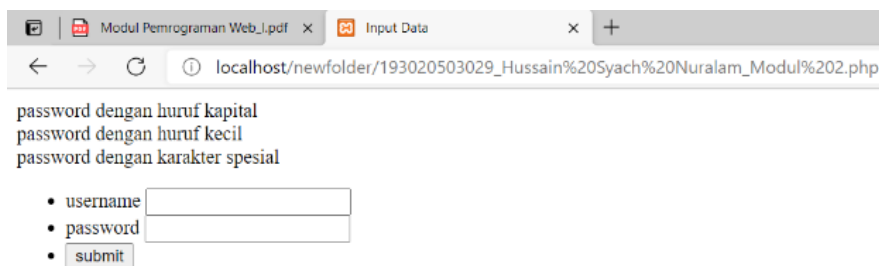
Gambar 2.4 Sintaks Percabangan Program Input Data Serta Penggunaan Handling

```

39         if( $x == false ){
40             echo "berhasil";
41         }
42     }
43
44     ?>

```

Gambar 2.5 Sintaks Percabangan Program Input Data Serta Penggunaan Handling



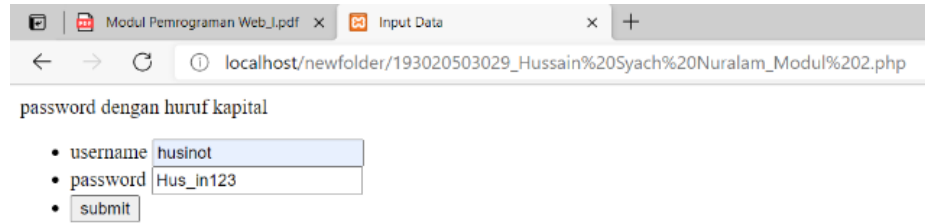
Modul Pemrograman Web_Lpdf x Input Data x +

localhost/newfolder/193020503029_Hussain%20Syach%20Nuralam_Modul%202.php

password dengan huruf kapital
password dengan huruf kecil
password dengan karakter spesial

- username
- password
-

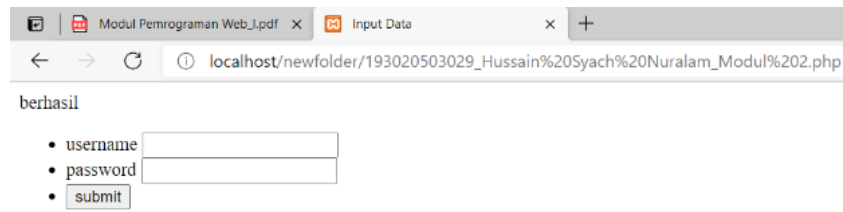
Gambar 2.6 Program Gagal Submit Karena Tidak Sesuai



The screenshot shows a web browser window with two tabs: 'Modul Pemrograman Web_I.pdf' and 'Input Data'. The address bar displays 'localhost/newfolder/193020503029_Hussain%20Syach%20Nuralam_Modul%202.php'. Below the address bar, the text 'password dengan huruf kapital' is visible. The form consists of three items: a bullet point followed by 'username' and a text input field containing 'husinot'; a bullet point followed by 'password' and a text input field containing 'Hus_in123'; and a bullet point followed by a 'submit' button.

- username
- password
-

Gambar 2.7 Pengisian Data Username dan Password



The screenshot shows the same web browser window as in Gambar 2.7. The text 'berhasil' is now displayed above the form fields. The form fields are empty, and the 'submit' button is still present.

- username
- password
-

Gambar 2.8 Program Berhasil Submit